

Komunikace mezi serverem a roboty je realizována plně textovým protokolem. Každý příkaz je zakončen dvojicí speciálních symbolů „\a\b“. (Jsou to **dva** znaky 'a' a 'b'.) Server musí dodržet komunikační protokol do detailu přesně, ale musí počítat s nedokonalými firmwary robotů (viz sekce Speciální situace).

Zprávy serveru:

| Název | Zpráva | Popis |
|-------------------------------|---|---|
| SERVER_CONFIRMATION | <16-bitové číslo v decimální notaci>\a\b | Zpráva s potvrzovacím kódem. Může obsahovat maximálně 5 čísel a ukončovací sekvenci \a\b. |
| SERVER_MOVE | 102 MOVE\a\b | Příkaz pro pohyb o jedno pole vpřed |
| SERVER_TURN_LEFT | 103 TURN LEFT\a\b | Příkaz pro otočení doleva |
| SERVER_TURN_RIGHT | 104 TURN RIGHT\a\b | Příkaz pro otočení doprava |
| SERVER_PICK_UP | 105 GET MESSAGE\a\b | Příkaz pro vyzvednutí zprávy |
| SERVER_LOGOUT | 106 LOGOUT\a\b | Příkaz pro ukončení spojení po úspěšném vyzvednutí zprávy |
| SERVER_KEY_REQUEST | 107 KEY REQUEST\a\b | Žádost serveru o Key ID pro komunikaci |
| SERVER_OK | 200 OK\a\b | Kladné potvrzení |
| SERVER_LOGIN_FAILED | 300 LOGIN FAILED\a\b | Nezdařená autentizace |
| SERVER_SYNTAX_ERROR | 301 SYNTAX ERROR\a\b | Chybná syntaxe zprávy |
| SERVER_LOGIC_ERROR | 302 LOGIC ERROR\a\b | Zpráva odeslaná ve špatné situaci |
| SERVER_KEY_OUT_OF_RANGE_ERROR | 303 KEY OUT OF RANGE\a\b | Key ID není v očekávaném rozsahu |

Zprávy klienta:

| Název | Zpráva | Popis | Ukázka | Maximální délka |
|---------------------|--|--|----------------|-----------------|
| CLIENT_USERNAME | <user name>\a\b | Zpráva s uživatelským jménem. Jméno může být libovolná sekvence znaků kromě dvojice \a\b. | Umpa_Lumpa\a\b | 20 |
| CLIENT_KEY_ID | <Key ID>\a\b | Zpráva obsahující Key ID. Může obsahovat pouze celé číslo o maximálně třech cifrách. | 2\a\b | 5 |
| CLIENT_CONFIRMATION | <16-bitové číslo v decimální notaci>\a\b | Zpráva s potvrzovacím kódem. Může obsahovat maximálně 5 čísel a ukončovací sekvenci \a\b. | 1009\a\b | 7 |
| CLIENT_OK | OK <x> <y>\a\b | Potvrzení o provedení pohybu, kde x a y jsou souřadnice robota po provedení pohybového příkazu. | OK -3 -1\a\b | 12 |
| CLIENT_RECHARGING | RECHARGING\a\b | Robot se začal dobíjet a přestal reagovat na zprávy. | | 12 |
| CLIENT_FULL_POWER | FULL POWER\a\b | Robot doplnil energii a opět přijímá příkazy. | | 12 |
| CLIENT_MESSAGE | <text>\a\b | Text vyzvednutého tajného vzkazu. Může obsahovat jakékoliv znaky kromě ukončovací sekvence \a\b. | Haf!\a\b | 100 |

Časové konstanty:

| Název | Hodnota [s] | Popis |
|--------------------|-------------|--|
| TIMEOUT | 1 | Server i klient očekávají od protistrany odpověď po dobu tohoto intervalu. |
| TIMEOUT_RECHARGING | 5 | Časový interval, během kterého musí robot dokončit dobíjení. |

Autentizace

Server i klient oba znají pět dvojic autentizačních klíčů (nejedná se o veřejný a soukromý klíč):

| Key ID | Server Key | Client Key |
|--------|------------|------------|
| 0 | 23019 | 32037 |
| 1 | 32037 | 29295 |
| 2 | 18789 | 13603 |
| 3 | 16443 | 29533 |
| 4 | 18189 | 21952 |

Každý robot začne komunikaci odesláním svého uživatelského jména (zpráva CLIENT_USERNAME). Uživatelské jméno může být libovolná sekvence 18 znaků neobsahující sekvenci „\a\b“. V dalším kroku vyzve server klienta k odeslání Key ID (zpráva SERVER_KEY_REQUEST), což je vlastně identifikátor dvojice klíčů, které chce použít pro autentizaci. Klient odpoví zprávou CLIENT_KEY_ID, ve které odešle Key ID. Po té server zná správnou dvojici klíčů, takže může spočítat "hash" kód z uživatelského jména podle následujícího vzorce:

Uživatelské jméno: Mnau!

ASCII reprezentace: 77 110 97 117 33

Výsledný hash: $((77 + 110 + 97 + 117 + 33) * 1000) \% 65536 = 40784$

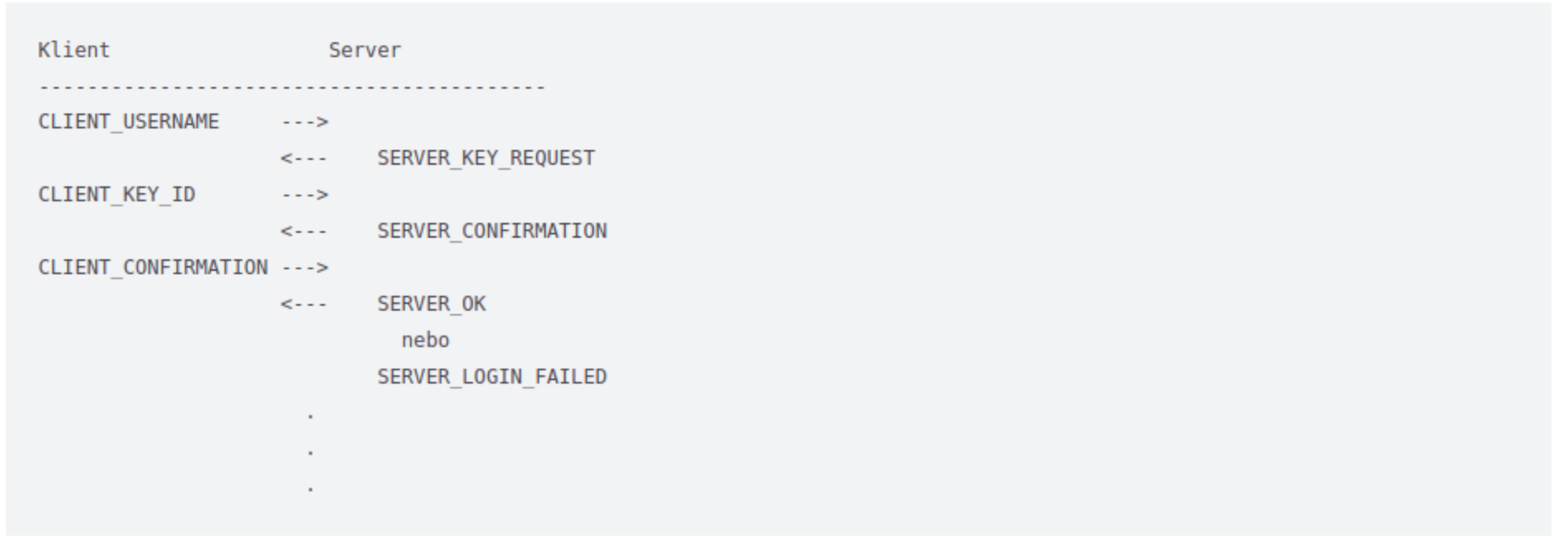
Výsledný hash je 16-bitové číslo v decimální podobě. Server poté k hashi přičte klíč serveru tak, že pokud dojde k překročení kapacity 16-bitů, hodnota jednoduše přetečení:

$$(40784 + 54621) \% 65536 = 29869$$

Výsledný potvrzovací kód serveru se jako text pošle klientovi ve zprávě SERVER_CONFIRM. Klient z obdrženého kódu vypočítá zpátky hash a porovná ho s očekávaným hashem, který si sám spočítal z uživatelského jména. Pokud se shodují, vytvoří potvrzovací kód klienta. Výpočet potvrzovacího kódu klienta je obdobný jako u serveru, jen se použije klíč klienta:

$$(40784 + 45328) \% 65536 = 20576$$

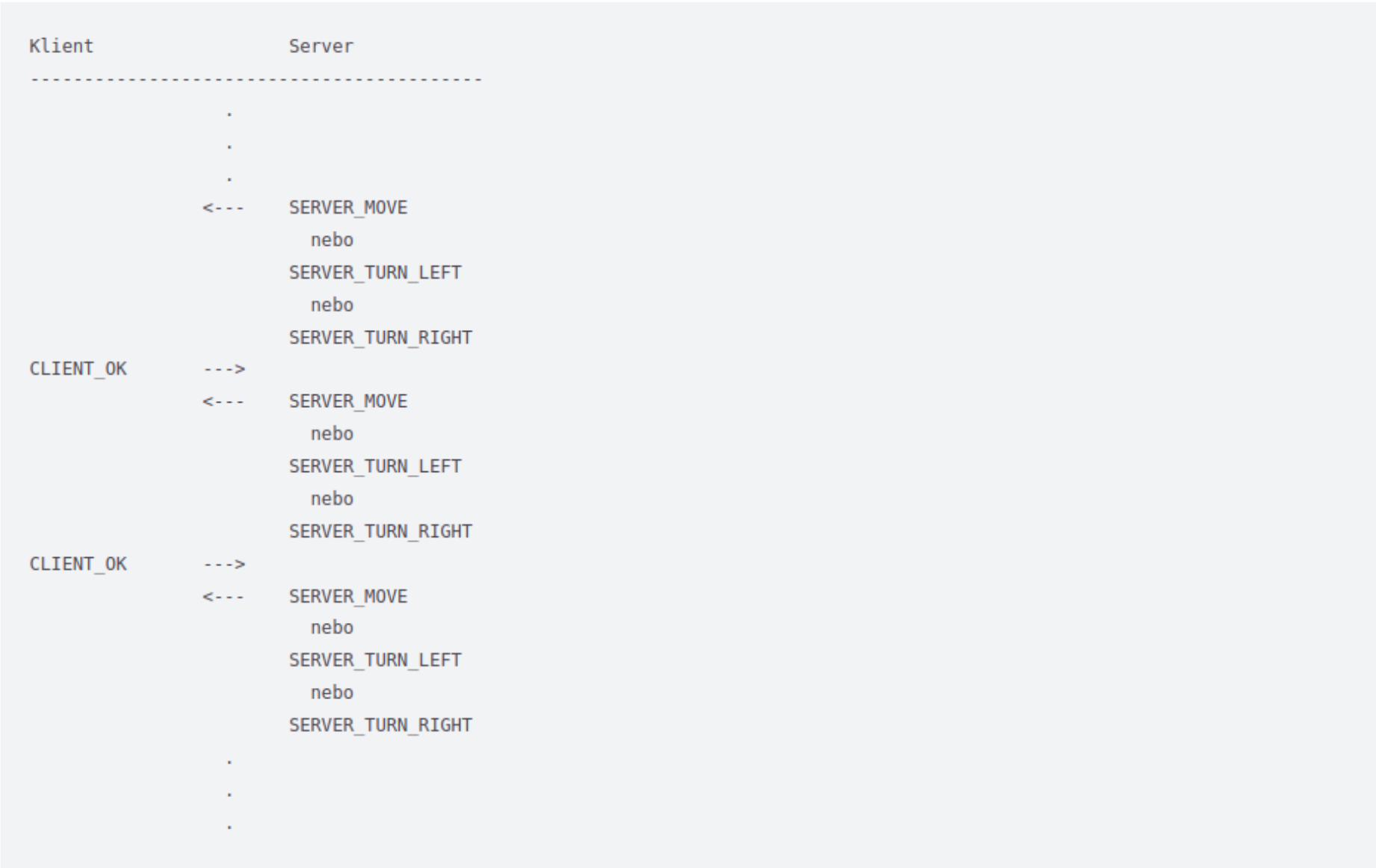
Potvrzovací kód klienta se odešle serveru ve zprávě CLIENT_CONFIRMATION, který z něj vypočítá zpátky hash a porovná jej s původním hashem uživatelského jména. Pokud se obě hodnoty shodují, odešle zprávu SERVER_OK, v opačném případě reaguje zprávou SERVER_LOGIN_FAILED a ukončí spojení. Celá sekvence je na následujícím obrázku:



Server dopředu nezná uživatelská jména. Roboti si proto mohou zvolit jakékoliv jméno, ale musí znát sadu klíčů klienta i serveru. Dvojice klíčů zajistí oboustranou autentizaci a zároveň zabrání, aby byl autentizační proces kompromitován prostým odposlechem komunikace.

Pohyb robota k cíli

Robot se může pohybovat pouze rovně (SERVER_MOVE) a je schopen provést otočení na místě doprava (SERVER_TURN_RIGHT) i doleva (SERVER_TURN_LEFT). Po každém příkazu k pohybu odešle potvrzení (CLIENT_OK), jehož součástí je i aktuální souřadnice. Pozice robota není serveru na začátku komunikace známa. Server musí zjistit polohu robota (pozici a směr) pouze z jeho odpovědí. Z důvodů prevence proti nekonečnému bloudění robota v prostoru, má každý robot omezený počet pohybů (pouze posunutí vpřed). Počet pohybů by měl být dostatečný pro rozumný přesun robota k cíli. Následuje ukázka komunikace. Server nejdříve pohne dvakrát robotem kupředu, aby detekoval jeho aktuální stav a po té jej navádí směrem k cílové souřadnici [0,0].



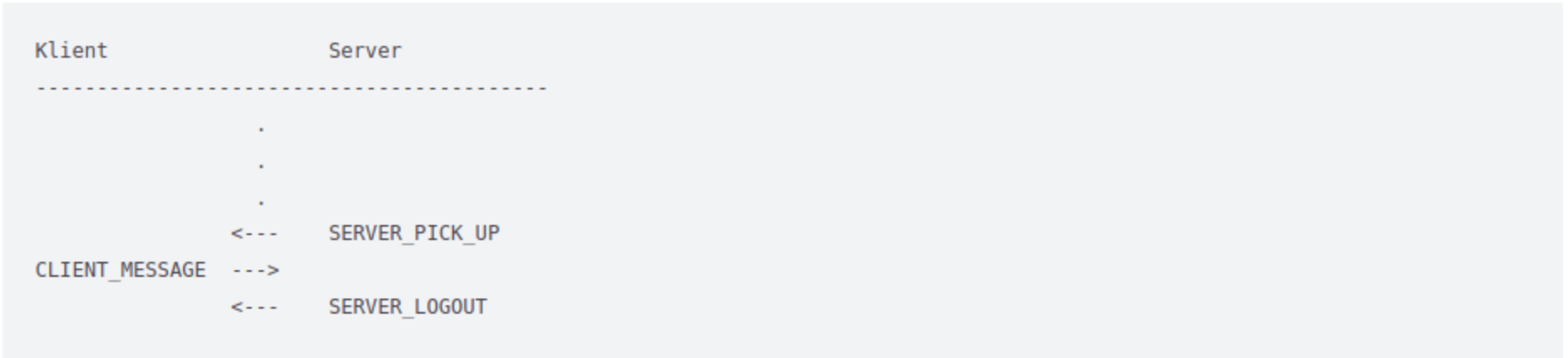
Těsně po autentizaci robot očekává alespoň jeden pohybový příkaz - SERVER_MOVE, SERVER_TURN_LEFT nebo SERVER_TURN_RIGHT! Nelze rovnou zkoušet vyzvednout tajemství. Po cestě k cíli se nachází mnoho překážek, které musí roboti překonat objíždkou. Pro překážky platí následující pravidla:

- Překážka okupuje vždy jedinou souřadnici.
- Je zaručeno, že každá překážka má prázdné všechny sousední souřadnice (tedy vždy lze jednoduše objet).
- Je zaručeno, že překážka nikdy neokupuje souřadnici [0,0].
- Pokud robot narazí do překážky více než dvacetkrát, poškodí se a ukončí spojení.

Překážka je detekována tak, že robot dostane pokyn pro pohyb vpřed (SERVER_MOVE), ale nedojde ke změně souřadnic (zpráva CLIENT_OK obsahuje stejné souřadnice jako v předchozím kroku). Pokud se pohyb neprovede, nedojde k odečtení z počtu zbývajících kroků robota.

Vyzvednutí tajného vzkazu

Poté, co robot dosáhne cílové souřadnice [0,0], tak se pokusí vyzvednout tajný vzkaz (zpráva SERVER_PICK_UP). Pokud je robot požádán o vyzvednutí vzkazu a nenachází se na cílové souřadnici, spustí se autodestrukce robota a komunikace se serverem je přerušena. Při pokusu o vyzvednutí na cílově souřadnici reaguje robot zprávou CLIENT_MESSAGE. Server musí na tuto zprávu zareagovat zprávou SERVER_LOGOUT. (Je zaručeno, že tajný vzkaz se nikdy neshoduje se zprávou CLIENT_RECHARGING, pokud je tato zpráva serverem obdržena po žádosti o vyzvednutí jedná se vždy o dobíjení.) Poté klient i server ukončí spojení. Ukázka komunikace s vyzvednutím vzkazu:



Dobíjení

Každý z robotů má omezený zdroj energie. Pokud mu začne docházet baterie, oznámí to serveru a poté se začne sám ze solárního panelu dobíjet. Během dobíjení nereaguje na žádné zprávy. Až skončí, informuje server a pokračuje v činnosti tam, kde přestal před dobíjením. Pokud robot neukončí dobíjení do časového intervalu TIMEOUT_RECHARGING, server ukončí spojení.



Další ukázka:



Chybové situace

Někteří roboti mohou mít poškozený firmware a tak mohou komunikovat špatně. Server by měl toto nevhodné chování detekovat a správně zareagovat.

Chyby při autentizaci

Pokud je ve zprávě CLIENT_KEY_ID Key ID, který je mimo očekávaný rozsah (tedy číslo, které není mezi 0-4), tak na to server reaguje chybovou zprávou SERVER_KEY_OUT_OF_RANGE_ERROR a ukončí spojení. Za číslo se pro zjednodušení považují i záporné hodnoty. Pokud ve zprávě CLIENT_KEY_ID není číslo (např. písmena), tak na to server reaguje chybou SERVER_SYNTAX_ERROR.

Pokud je ve zprávě CLIENT_CONFIRMATION číselná hodnota (i záporné číslo), která neodpovídá očekávanému potvrzení, tak server pošle zprávu SERVER_LOGIN_FAILED a ukončí spojení. Pokud se nejedná vůbec o číste číselnou hodnotu, tak server pošle zprávu SERVER_SYNTAX_ERROR a ukončí spojení.

Syntaktická chyba

Na syntaktickou chybu reaguje server vždy okamžitě po obdržení zprávy, ve které chybu detekoval. Server pošle robotovi zprávu SERVER_SYNTAX_ERROR a pak musí co nejdříve ukončit spojení. Syntakticky nekorektní zprávy:

- Příchozí zpráva je delší než počet znaků definovaný pro každou zprávu (včetně ukončovacích znaků \a\b). Délky zpráv jsou definovány v tabulce s přehledem zpráv od klienta.
- Příchozí zpráva syntakticky neodpovídá ani jedné ze zpráv CLIENT_USERNAME, CLIENT_KEY_ID, CLIENT_CONFIRMATION, CLIENT_OK, CLIENT_RECHARGING a CLIENT_FULL_POWER.

Každá příchozí zpráva je testována na maximální velikost a pouze zprávy CLIENT_CONFIRMATION, CLIENT_OK, CLIENT_RECHARGING a CLIENT_FULL_POWER jsou testovány na jejich obsah (zprávy CLIENT_USERNAME a CLIENT_MESSAGE mohou obsahovat cokoliv).

Logická chyba

Logická chyba nastane pouze při nabíjení - když robot pošle info o dobíjení (CLIENT_RECHARGING) a po té pošle jakoukoliv jinou zprávu než CLIENT_FULL_POWER nebo pokud pošle zprávu CLIENT_FULL_POWER, bez předchozího odeslání CLIENT_RECHARGING. Server na takové situace reaguje odesláním zprávy SERVER_LOGIC_ERROR a okamžitým ukončením spojení.

Timeout

Protokol pro komunikaci s roboty obsahuje dva typy timeoutu:

- TIMEOUT - timeout pro komunikaci. Pokud robot nebo server neobdrží od své protistrany žádnou komunikaci (nemusí to být však celá zpráva) po dobu tohoto časového intervalu, považují spojení za ztracené a okamžitě ho ukončí.
- TIMEOUT_RECHARGING - timeout pro dobíjení robota. Po té, co server přijme zprávu CLIENT_RECHARGING, musí robot nejpozději do tohoto časového intervalu odeslat zprávu CLIENT_FULL_POWER. Pokud to robot nestihne, server musí okamžitě ukončit spojení.

Speciální situace

Při komunikaci přes komplikovanější síťovou infrastrukturu může docházet ke dvěma situacím:

- Zpráva může dorazit rozdělena na několik částí, které jsou ze socketu čteny postupně. (K tomu dochází kvůli segmentaci a případnému zdržení některých segmentů při cestě sítí.)
- Zprávy odeslané brzy po sobě mohou dorazit téměř současně. Při jednom čtení ze socketu mohou být načteny obě najednou. (Tohle se stane, když server nestihne z bufferu načíst první zprávu dříve než dorazí zpráva druhá.)

Za použití přímého spojení mezi serverem a roboty v kombinaci s výkonným hardwarem nemůže k těmto situacím dojít přirozeně, takže jsou testovačem vytvářeny uměle. V některých testech jsou obě situace kombinovány.

Každý správně implementovaný server by se měl umět s touto situací vyrovnat. Firmware robotů s tímto faktem počítají a dokonce ho rády zneužívají.

Pokud se v protokolu vyskytuje situace, kdy mají zprávy od robota předem dané pořadí, jsou v tomto pořadí odeslány najednou. To umožňuje sondám snížit jejich spotřebu a zjednodušuje to implementaci protokolu (z jejich pohledu).

Optimalizace serveru

Server optimalizuje protokol tak, že nečeká na dokončení zprávy, která je očividně špatná. Například na výzvu k autentizaci pošle robot pouze část zprávy s uživatelským jménem. Server obdrží např. 22 znaků uživatelského jména, ale stále neobdržel ukončovací sekvenci \al\b. Vzhledem k tomu, že maximální délka zprávy je 20 znaků, je jasné, že přijímaná zpráva nemůže být validní. Server tedy zareaguje tak, že nečeká na zbytek zprávy, ale pošle zprávu `SERVER_SYNTAX_ERROR` a ukončí spojení. V principu by měl postupovat stejně při vyzvedávání tajného vzkazu.

V případě části komunikace, ve které se robot naviguje k cílovým souřadnicím očekává tři možné zprávy: `CLIENT_OK`, `CLIENT_RECHARGING` nebo `CLIENT_FULL_POWER`. Pokud server načte část neúplné zprávy a tato část je delší než maximální délka těchto zpráv, pošle `SERVER_SYNTAX_ERROR` a ukončí spojení. Pro pomoc při optimalizaci je u každé zprávy v tabulce uvedena její maximální velikost.

Ukázka komunikace

```
C: "0ompa Loompa\a\b"  
S: "107 KEY REQUEST\a\b"  
C: "0\a\b"  
S: "64907\a\b"  
C: "8389\a\b"  
S: "200 OK\a\b"  
S: "102 MOVE\a\b"  
C: "OK 0 0\a\b"  
S: "102 MOVE\a\b"  
C: "OK -1 0\a\b"  
S: "104 TURN RIGHT\a\b"  
C: "OK -1 0\a\b"  
S: "104 TURN RIGHT\a\b"  
C: "OK -1 0\a\b"  
S: "102 MOVE\a\b"  
C: "OK 0 0\a\b"  
S: "105 GET MESSAGE\a\b"  
C: "Tajny vzkaz.\a\b"  
S: "106 LOGOUT\a\b"
```